

Fig. 1.

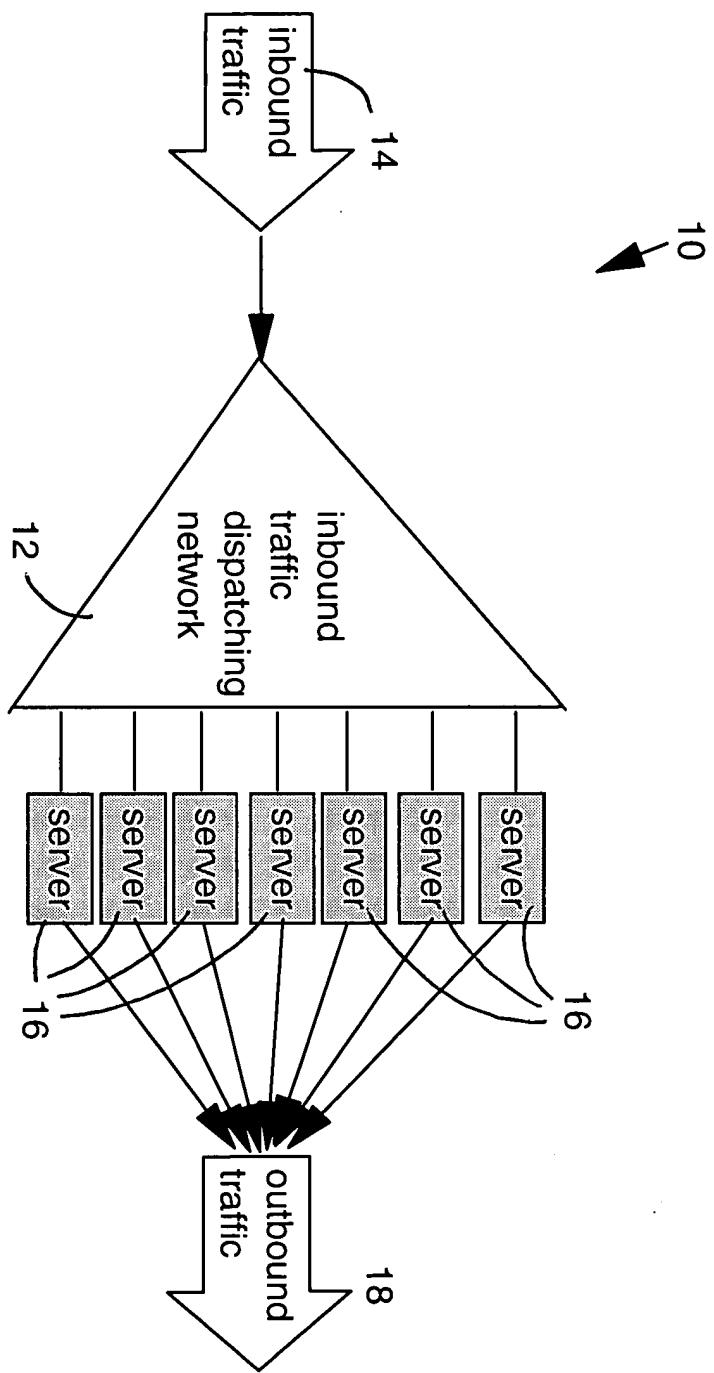


Fig. 2.

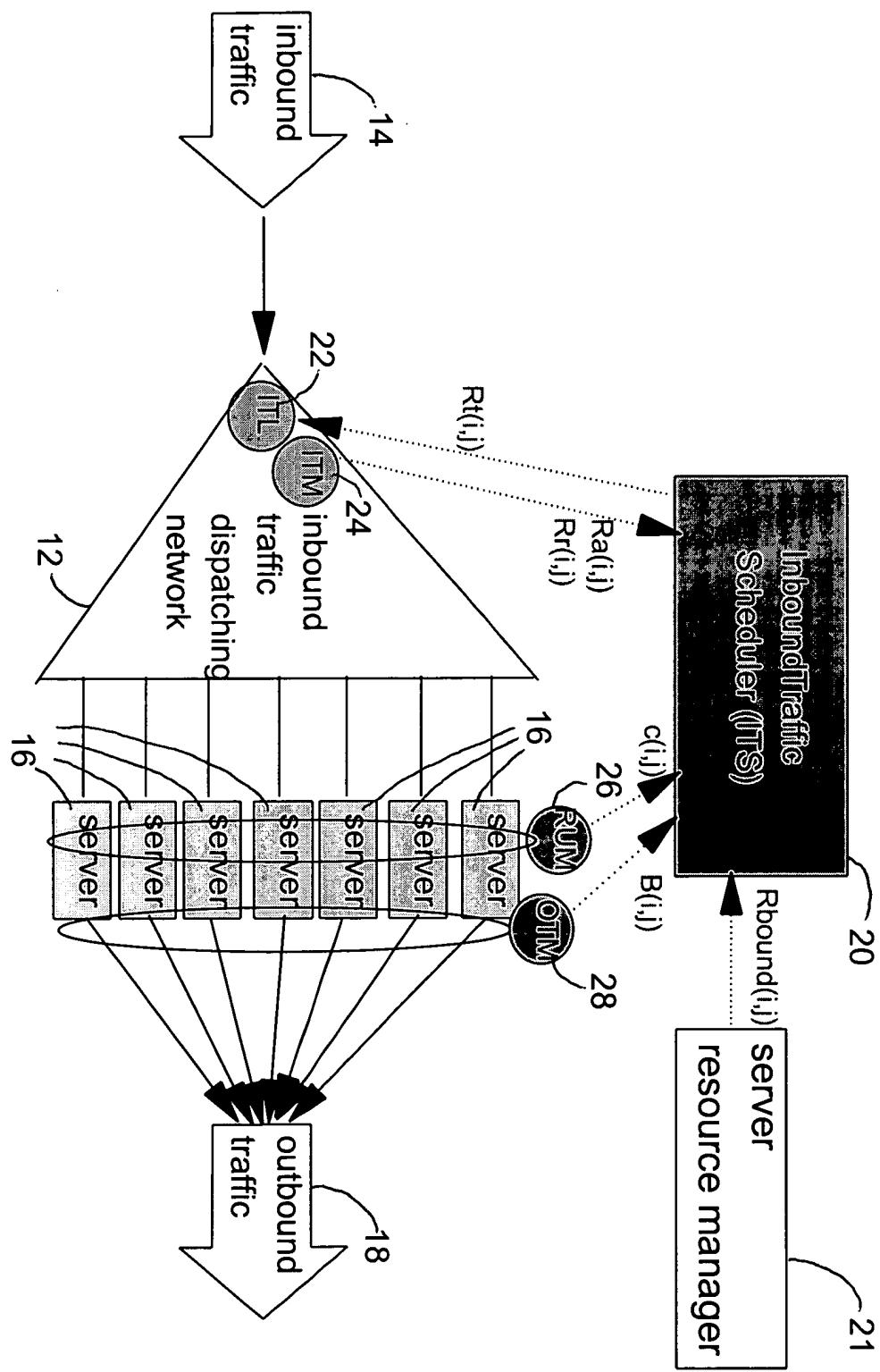


Fig. 3

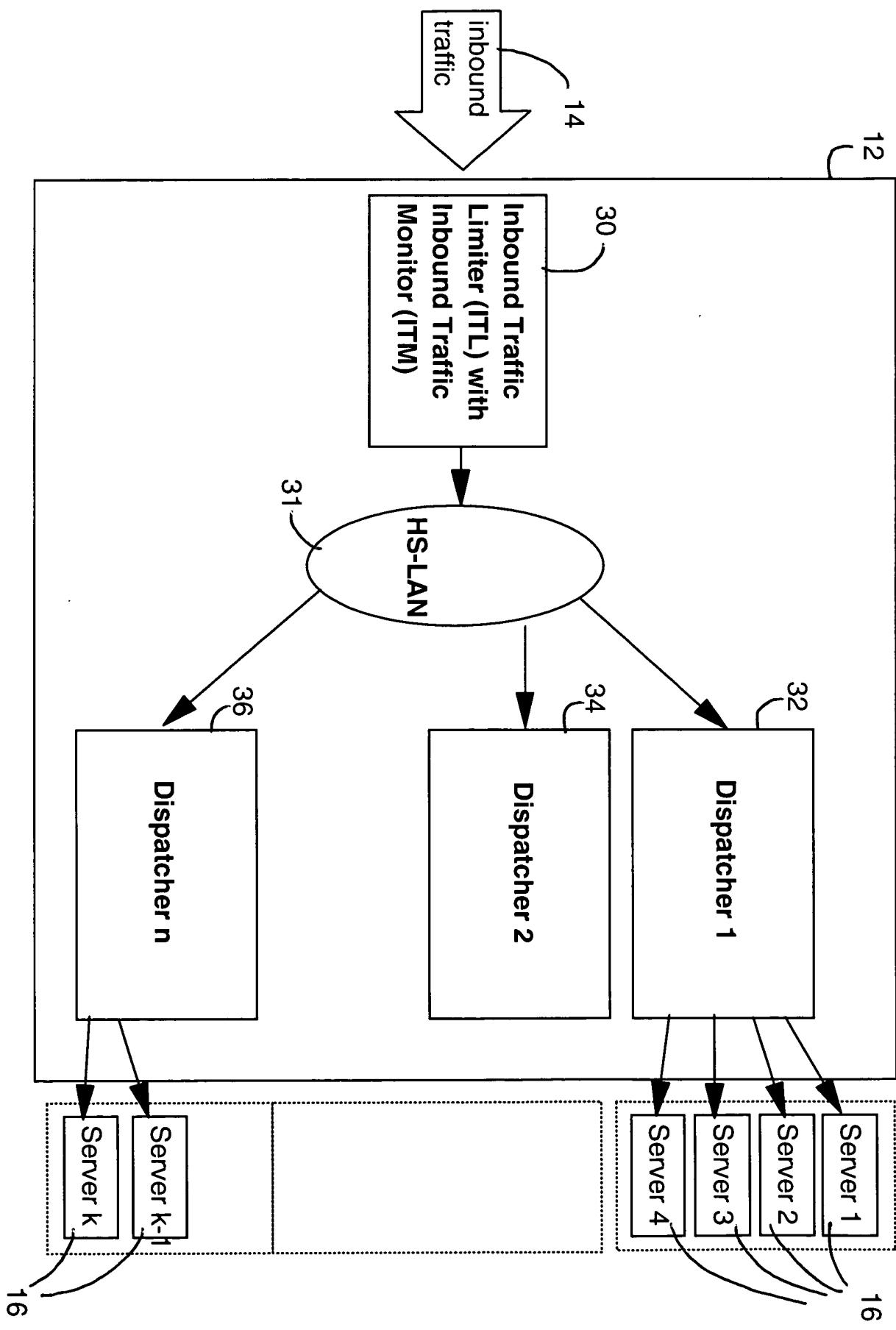
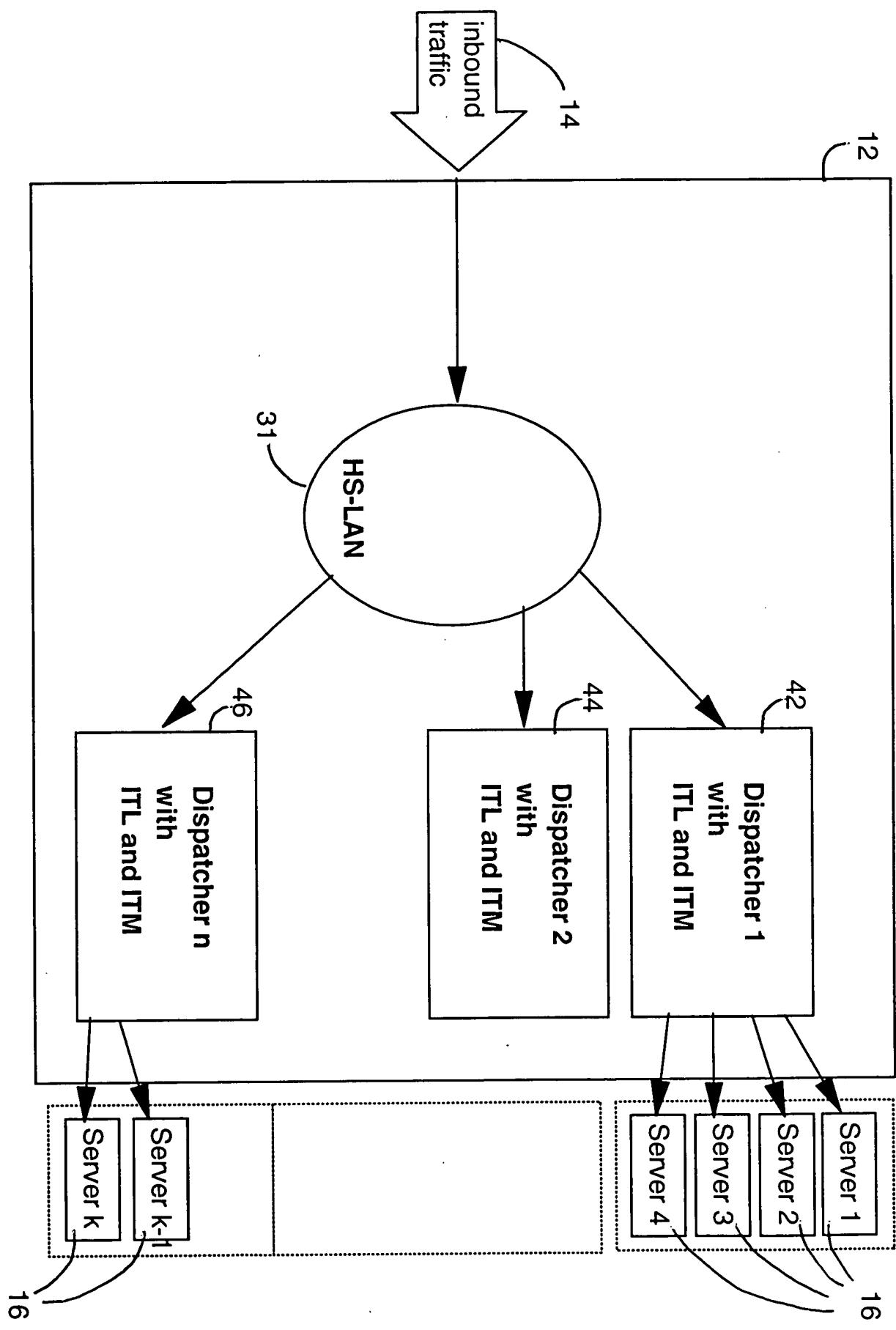


Fig. 4



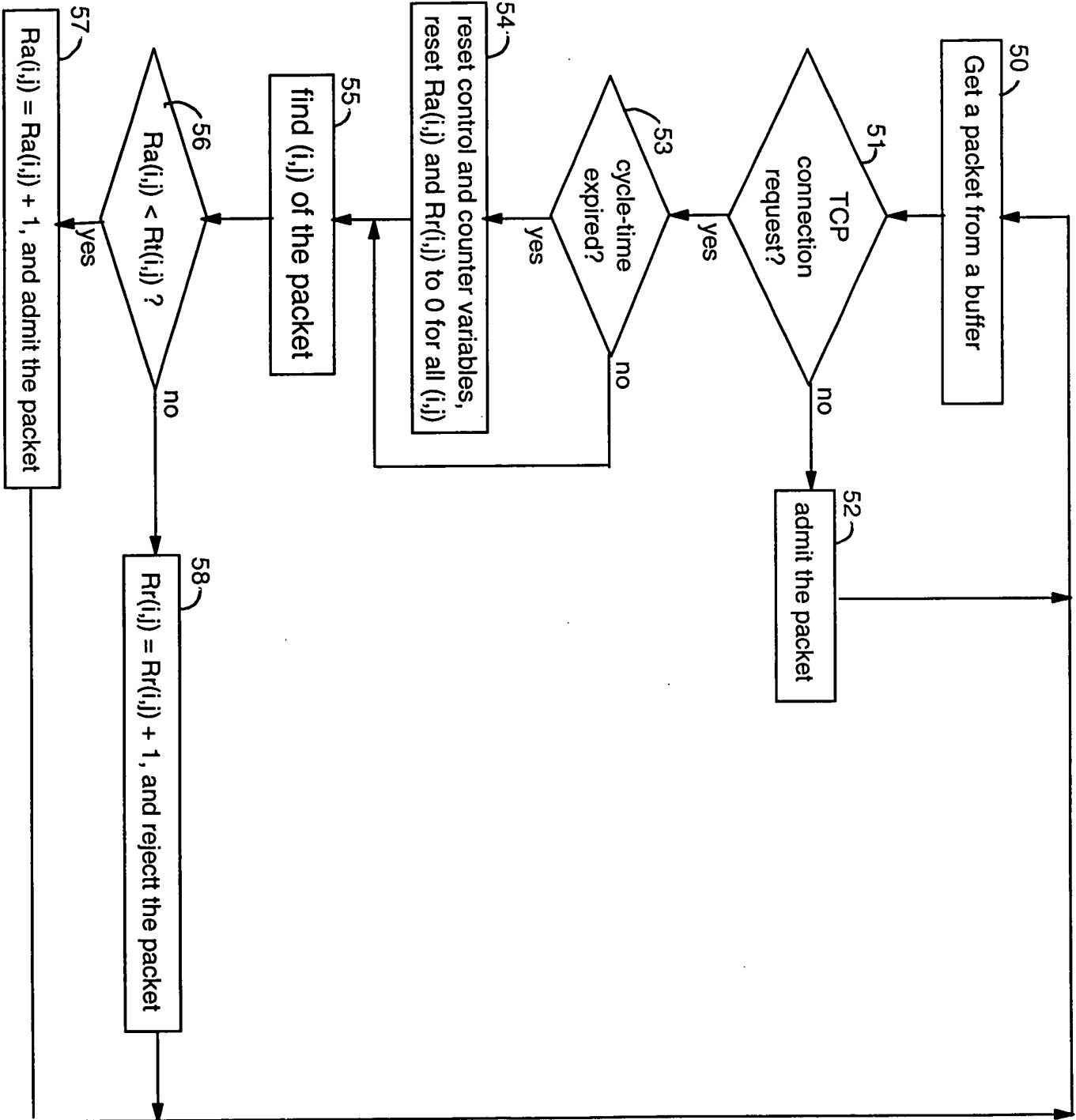


Fig. 5

**Step 1**

```

START:
For all (i,j), set Bmax(i,j) = min (Bmax(i,j),Rbound(i,j)*b(i,j)); /* reflecting "external constraint" Bbound(i,j) */
For all (i,j), set Bt(i,j) = b(i,j)*Ra(i,j); /* estimating the current outbound traffic */
Let Bt be the sum of Bt(i,j) over all (i,j); /* estimating the current total outbound traffic */
If Bt>Btotal then go to step 2; /* link congestion detected */
  /* no request rejection */
  /* SLA is not violated. "<=" means less-than-or-equal-to */
  If ((R(i,j)>0) &(Bt(i,j)<=Bmax(i,j))) over all (i,j)
    then go to step 5; else go to step2;

```

**Step 2**

```

COMPUTE_BANDWIDTH_TARGETS: /* computing new targets for bandwidth usage */
For all (i,j), set Bt(i,j) = b(i,j)*(Ra(i,j)+R(i,j)); /* estimating outbound traffic when all requests are admitted */
Let Bt be the sum of Bt(i,j) for all (i,j); /* this step is needed since Bt(i,j) were just re-computed */
For every (i,j) such that Bt(i,j)>Bmax(i,j) /* wants to generate more than the maximum SLA */
  first set Bt = Bt-(Bt(i,j)-Bmax(i,j)) /* adjusting expected total outbound traffic */
  and then set Bt(i,j) = Bmax(i,j); /* bounding traffic by maximum SLA */
  /* no link congestion will be anticipated */
If Bt<=Btotal
  then go to step 1; else go to step 3;

```

**Step 3**

```

Let Bexcess be the sum of (Bt(i,j)-Bmin(i,j)) over those Bt(i,j)>Bmin(i,j); /* computing "excess" bandwidth */
/* perform either Case 1 or Case 2 */
/* Case 1: compute "sharable" bandwidth when bandwidth borrowing is permitted */
Let Bsharable be Btotal minus the sum of smaller of (Bt(i,j) and Bmin(i,j)) over all (i,j);
/* Case 2: compute "sharable bandwidth when bandwidth borrowing is not permitted */
Let Bsharable be Btotal minus the sum of Bmin(i,j) over all (i,j);
For every (i,j) such that Bt(i,j)>Bmin(i,j) /* perform fair proration */
  set Bt(i,j) = Bmin(i,j) + (Bt(i,j)-Bmin(i,j))*(Bsharable / Bexcess);

```

**Step 4**

```

COMPUTE_NEW_RATES:
For every (i,j) such that Bt(i,j)<=Bmin(i,j) set Bt(i,j) = Bmax(i,j); /* this is equivalent to "no throttling" */
For every (i,j) set R(i,j) = Bt(i,j) / b(i,j); /* computing target rates */
  Optionally compute R(i,j,k) from R(i,j) for all k; /* optional computation, rate for each server */

```

**Step 5**

STOP:

62

63

64

65

Fig. 7

